Analisi del Codice Assembly di un Malware

Il codice assembly fornito è parte di un campione di malware che verifica la presenza di una connessione internet attiva. Ecco una spiegazione dettagliata del codice e della sua funzionalità:

Analisi del Codice Assembly

```assembly
.text:00401000    push    ebp
.text:00401001    mov     ebp, esp
.text:00401003    push    ecx
.text:00401004    push    0               ; dwReserved
.text:00401006    push    0               ; lpdwFlags
.text:00401008    call    ds:InternetGetConnectedState
.text:0040100E    mov     [ebp+var_4], eax
.text:00401011    cmp     [ebp+var_4], 0
.text:00401015    jz      short loc_40102B
.text:00401017    push    offset aSuccessInterne ; "Success: Internet Connection
"
.text:0040101C    call    sub_40105F
.text:00401021    add     esp, 4
.text:00401024    mov     eax, 1
.text:00401029    jmp     short loc_40103A
.text:0040102B    ; code continues
```

Costrutti Noti e Spiegazione

1. **Prologo ed Epilogo della Funzione:**
   - Il codice inizia con la configurazione del frame dello stack usando `push ebp` e `mov ebp, esp`. Questo è un prologo comune per le funzioni in assembly.

2. **Chiamata a `InternetGetConnectedState`:**
   - La funzione `InternetGetConnectedState` è chiamata con due parametri passati sullo stack, entrambi impostati a 0. Questa funzione controlla se la macchina ha una connessione internet attiva.
   - `push 0` due volte prima della chiamata corrisponde ai parametri `lpdwFlags` e `dwReserved`, entrambi impostati a 0.

3. **Memorizzazione e Verifica del Risultato:**
   - Il risultato di `InternetGetConnectedState` è memorizzato in `eax`, che viene poi spostato in `[ebp+var_4]`.
   - Il codice confronta il risultato con 0 usando `cmp` e `jz`. Se il risultato è zero, significa che non c'è connessione internet, e il codice salta a `loc_40102B`.

4. **Percorso di Successo:**
   - Se c'è una connessione internet (cioè `eax` non è zero), viene spinto l'indirizzo della stringa "Success: Internet Connection
" e viene chiamata una subroutine `sub_40105F`, che probabilmente si occupa di visualizzare o registrare il messaggio.
   - Dopo la chiamata alla subroutine, lo stack viene pulito aggiungendo 4 a `esp`, e `eax` è impostato a 1 (indicando successo).

5. **Salto alla Prossima Istruzione:**
    - Infine, il codice salta a `loc_40103A`, bypassando il codice di gestione del fallimento in `loc_40102B`.

Funzionalità

Il codice assembly fornito implementa una funzione che verifica la presenza di una connessione internet attiva e registra un messaggio di successo se la connessione è presente. Ecco il pseudocodice per una migliore comprensione:

```c
void checkInternetConnection() {
    int connectionStatus = InternetGetConnectedState(0, 0);
    if (connectionStatus != 0) {
        log("Success: Internet Connection\n");
        return 1;
    } else {
        // Gestione del caso di assenza di connessione (codice non mostrato)
    }
}
```

- `InternetGetConnectedState(0, 0)` è usato per determinare se la macchina ha accesso a internet.
- Se il risultato è diverso da zero, registra "Success: Internet Connection\n" e ritorna 1.
- Se il risultato è zero, salta a una diversa parte del codice per gestire l'assenza di una connessione

internet.

Conclusione

Questo pezzo di codice assembly è responsabile per rilevare se la macchina host ha una connessione internet attiva e registra un messaggio di successo se la connessione è presente. Questo è tipicamente parte della routine di controllo della rete di un malware per assicurarsi che possa comunicare con i server di comando e controllo o scaricare payload aggiuntivi.